CS 4504 – Computer Architecture
Prof. Kirk W. Cameron
Fall 2006 Project 3: Using the Simplescalar Simulator

Important Dates

Assigned: April 4, 2006
Official due date: April 20, 2006
Extended due date: April 25, 2006

Summary

In this project you are expected to use a portion of the Simplescalar Toolset to measure the performance of 3 or more benchmark codes under changes to the microarchitecture simulation. The goal is for you to gain familiarity with Simplescalar enough to perform minor experimentation and experience the pros and cons to simulation of microarchitectures. Undergrads will accomplish the first two problems in this assignment.

Notes

The Simplescalar toolset is available on our local Unix systems under TBD, but if you wish to download the toolset from the Simplescalar website (www.simplescalar.com), you may do so with the usual caveats (less support from myself and the TA). As usual if you choose this road, we'll do our best to answer conceptual questions on other platforms.

The TA should be your first point of contact for this assignment as he is our resident Simplescalar expert. Lecture notes on Simplescalar are available at the class website.

You will deliver a short report (< 10 pages) illustrating your results under the detailed guidelines below.

The Assignment

In your report, you should provide discussion and results for each problem accomplished. In class we learned the value of measuring CPI (cycles per instruction). The inverse of CPI is IPC (instructions per cycle). Most architecture papers present results in IPC to show the achieved "instruction-level parallelism" of a code. You are expected to provide results of the first two problems' measured benchmarks in similar format. Problem 3 requires a similar format to results presented in project one, but with results taken from Simplescalar. Take a look at some of the Figures in Chapter 3 of your text (replicated at the end of this paper). The number and type of figures is up to you. Generally you are trying to explain trends in the data sets, so you should use appropriate figures.

- You MUST accomplish problems 1 and 2.

- Extra credit may be attempted only after you have finished the assignment (i.e. not in lieu of the assignment). The credit awarded will depend on the work and creativity. For example, something that requires simple measurements of the existing simplescalar implementation would warrant a small amount of extra credit, while modifying the simulator to obtain additional measurements or change the hardware implementation is more substantial. Please identify any extra credit studies in your paper. There will be no extensions to the deadline for extra credit.

Problem 1: Cache performance study

Solve Problem 5.10 of the textbook. Change the ranges for the L1 cache to 16 KB to 128 KB, and for the L2 cache to 256 KB to 2 MB.

You should collect results for three or more codes. The first two codes should be lat_mem_rd.c from lmbench, and one of your matrix multiply implementations. Keep in mind these simulations may take a while to complete. Start early!

Problem 2: Performance of TLB implementations

Solve Problem 5.20 of the textbook. Use a direct-mapped TLB in place of two-way set associative, and change the ranges for entries to 64, 128, 256, and 512.

You should collect results for three or more codes. The first two codes should be lat_mem_rd.c from lmbench, and one of your matrix multiply implementations. Keep in mind these simulations may take a while to complete. Start early!

NOTE: Compiling lat_mem_rd within Simplescalar will require modification of the lat_mem_rd.c source code. Basically, you will need to inline library calls, etc. You can write a version of the code from scratch if you prefer. Discuss the required modifications and methodologies (and difficulties) in your final report.